Named Data Networking in Climate Research and HEP Applications

2015 J. Phys.: Conf. Ser. 664 052033

(http://iopscience.iop.org/1742-6596/664/5/052033)

Journal of Physics: Conference Series **664** (2015) 052033 doi:10.1088/1742-6596/664/5/052033

Named Data Networking in Climate Research and HEP Applications

Susmit Shannigrahi[1], Christos Papadopoulos[1], Edmund Yeh[5], Harvey Newman[2], Artur Jerzy Barczyk[6], Ran Liu[6], Alex Sim[3], Azher Mughal[2], Inder Monga[4], Jean-Roch Vlimant[2], John Wu[3],

[1]Colorado State University, Fort Collins, CO 80521, USA
[2]California Institute of Technology, 1200 East California Boulevard, Pasadena, CA 91125, USA
[3]Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA
[4]ESNet, 1 Cyclotron Road, Berkeley, CA 94720, USA
[5]Northeastern University, 360 Huntington Avenue, Boston, MA 02115, USA
[6]Previously at California Institute of Technology (presently at Cisco Systems Inc), 1200 East California Boulevard, Pasadena, CA 91125, USA

E-mail: susmit@cs.colostate.edu, Artur.Barczyk@cern.ch, christos@cs.colostate.edu, asim@lbl.gov, imonga@es.net,vlimant@cern.ch, newman@hep.caltech.edu, azher@hep.caltech.edu, kwu@lbl.gov, eyeh@ece.neu.edu

Abstract.

The Computing Models of the LHC experiments continue to evolve from the simple hierarchical MONARC[2] model towards more agile models where data is exchanged among many Tier2 and Tier3 sites, relying on both large scale file transfers with strategic data placement, and an increased use of remote access to object collections with caching through CMS's AAA, ATLAS' FAX and ALICE's AliEn projects, for example. The challenges presented by expanding needs for CPU, storage and network capacity as well as rapid handling of large datasets of file and object collections have pointed the way towards future more agile pervasive models that make best use of highly distributed heterogeneous resources.

In this paper, we explore the use of Named Data Networking (NDN), a new Internet architecture focusing on content rather than the location of the data collections. As NDN has shown considerable promise in another data intensive field, Climate Science, we discuss the similarities and differences between the Climate and HEP use cases, along with specific issues HEP faces and will face during LHC Run2 and beyond, which NDN could address.

1. Introduction

Sciences such as Climate and High-Energy Physics impose significant challenges to data management. First, datasets are quite big, easily reaching into the TB and PB range; second, datasets can be distributed around the world; third, there may exist different versions, making it hard to locate the desired one; fourth, moving such datasets is not a trivial matter, it must be planned well ahead of time, it often requires operator intervention and management, and there is no guarantee that the full dataset will be retrieved from the closest location, or in some extreme cases – retrieved at all. For these reasons and more, data publishing, discovery and movement are very hard problems in these and other sciences.

Traditional IP networks do not provide an appropriate network service model to smoothly facilitate such operations. IP networks address hosts, where scientists typically care about the data. This host-centric property of IP networks poses several complications in managing scientific data: (a) the user must know a priori or use tools to discover where the desired dataset resides; (b) the user must implement robustness on top of the network service by creating multiple repositories and building appropriate mechanisms to synchronize them, detect failure and switch over to a nearby live repository; (c) to make delivery more efficient, the user must implement caching on top of the network and deploy distributed mechanisms to ensure cache coherency; (d) the user has no assurance that the dataset just obtained was not compromised, or is an older version; and more.

2. Named Data Networking

NDN, an instance of Information Centric Networking (ICN), is a new Internet architecture[3] which focuses on the what (the actual content), rather than the where (the host where the data resides). The NDN primitive has two types of packets, Interest and Data packets. Interest packets are used for asking questions and Data packers are used to send data back to the requester. The naming structure adopted by NDN has several desirable properties:

(i) it is an intuitive, common organizational structure (e.g., file systems, URLs, etc.) and often require minimal changes in the naming structure (ii) it is scalable (similar to hierarchical IP addresses), and (iii) it is coupled with longest prefix matching, which enables data discovery and enumeration.

For example, the following root dataset name is easily translated into a hierarchical NDN name.

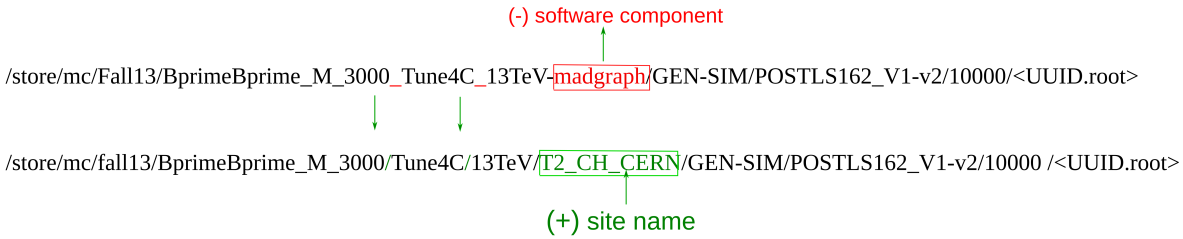

Figure 1. Naming a root dataset into NDN

The HEP community has, therefore, been able to agree on specific naming schemes and decide to keep the necessary name components, remove the ones not required or add additional components. In our example, we simply divided the original name in logical components separated with "/". We (arbitrarily) decided that the software-version component is not needed and can be replaced by a site name. This change, however, does not affect the underlying NDN layer as long as the names are hierarchical. To publish data, the data producer announces the whole name or a prefix of the name, e.g., /store/mc/fall13/ in the routing system. The intermediate routers receive these announcements and put them in the routing table. When users send out Interests for data ($/store/mc/fall13/../ < UUID.root >$), the Interests are forwarded toward the source of the routing advertisement. Once the producer receives the Interest, it sends back corresponding Data packets. Data is also cached along the return path, therefore anyone on the return path asking the same question receives a cached answer. This reduces latency and load on the original producer. Moreover, popular data can be strategically placed in nearby caches making them readily available for future requests. We discuss this in details in section6.

NDN adopts a drastically different security model than IP. Unlike traditional network security, which secures the data transmission channels, NDN secures the data by requiring that the producer digitally signs it. The receiver can now verify whether the data is valid or not. With digital signatures, provenance is easier to establish since the original publisher signature is verifiable regardless of where the data was retrieved. This has the added advantage that there is no need to secure the network intermediaries, and valid data can be retrieved even from a compromised host. In summary, NDN has a wide range of potential benefits such as in-network content caching with request deduplication to reduce congestion and improve delivery speed, simpler application configuration, and security built into the network at the data level.

3. NDN and Experience with the Climate Testbed

While some of the challenges are different between the climate and HEP domains, there are many similarities. These include high-speed transfer of large data collections, caching and replication, and effective management of the namespace. To better understand the challenges in climate applications, we have deployed a dedicated 6-node testbed for climate applications that reaches locations such as Atmospheric and Computer Science Departments at Colorado State University, LBNL and NWCS. We are in the process of adding several sites to the testbed, both inside and outside the US.

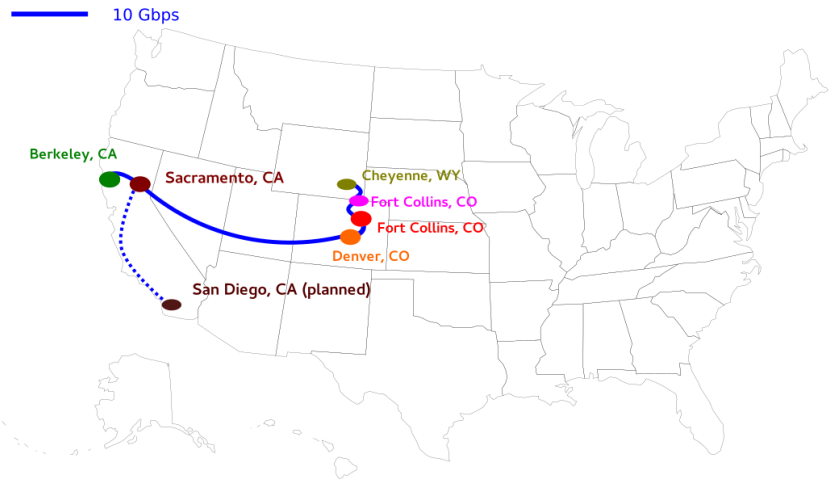

Figure 2. NDN testbed between ESNet, CSU and NWSC

The testbed was deployed with the support of ESnet. Connectivity is via 10G links with high-end machines that with 40 cores each, 128GB RAM and 48TB disk space. The machines cumulatively host over 50TB of climate data and are used for research, experimentation and development of climate applications. The machines are running Fedora 20/21 and have the latest NFD and ndn-cxx installed from upstream[6]. We have successfully begun translating names and testing NDN in the climate application domain [5]. To handle the various naming schemes used in climate applications we have designed and implemented translators that take existing names with arbitrary structure (generated by climate models, or home-grown) and translate them into NDN-compliant names. Depending on the original name structure, the translation can be fairly direct if data complies with the "Data Reference Syntax"[8]from the Coupled Model Intercomparison Project[4], or complex, such as home-grown naming schemes that require the analysis of metadata embedded in the dataset or even user feedback in order to construct proper NDN names. Even properly named datasets often present challenges: during our work with CMIP5 datasets we found some problems with the names, such as a mismatch between

name components in filename and metadata, missing components and data inconsistencies while translating the original names to NDN names.

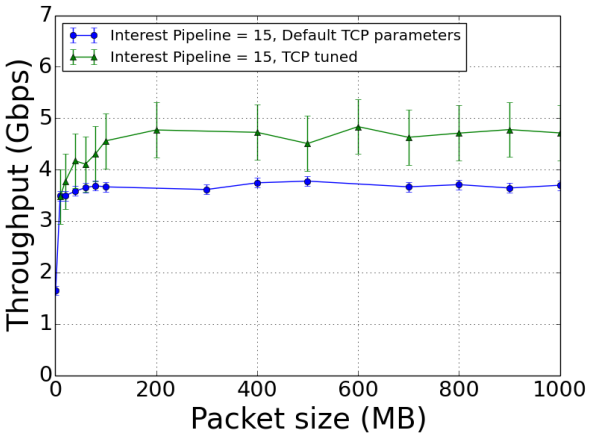

Figure 3. NFD and TCP Optimization for better throughput

High-speed transfers are an important requirement in the climate domain. To maximize the performance of the NDN forwarder (NFD), we have tuned the network stack extensively using ESnet's Linux tuning guide[7] . We also modified NFD and the ndn-cxx library to publish and receive large data objects. We found that tuning the TCP parameters along with large packets provides much better throughput than the default parameters provided by NFD and the Linux network stack.

We made the following optimizations:

(i) We set the packet size very large - we see that NFD throughput increases up to a packet size of 300MB. The default packet size of NFD is 8800 bytes. The latter is too small of a packet size to enable the high throughput required for scientific data.

(ii) The signing cost is expensive and in order to keep this from slowing down the transfer, we pre-sign the packets. We found this to be reasonable since most data needs to be signed during publication. This is a onetime operation and once data is published, there is no recurring cost. We note, however, that this probably won't apply to dynamically generated datasets and those operations would have to incur a one time packet signing cost.

(iii) We found that the default pipeline size of 4 Interests is too small for large data. Throughput continues to increase up to a pipeline size of 10 Interests. Beyond that, more pipelining does not speed up the transfer. More Interests provide effective transfer since NFD processed those Interests while data is being transmitted.

With the above optimizations, we were able to achieve around 5Gbps throughput with a packet size of 300MB and a pipeline of 10. The throughput is half of that achieved by the very mature TCP/IP stack at 9.5 Gbps. We expect the NFD throughput to approach TCP/IP once the software matures.

We are also trying to integrate OSCARS [1] with the NDN API on the testbed. We hope that OSCARS will provide us with capability to provision on-demand dedicated channels. This will be particularly helpful for critical data transfers since they will have their own reserved path and bandwidth.

The climate testbed has provided us with new insights into climate data, its naming conventions and the data management problems faced by climate scientists. In addition, it

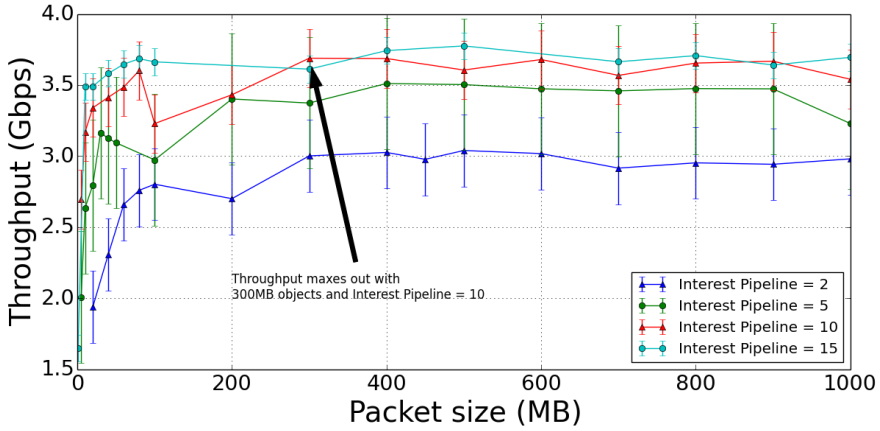

Figure 4. Pipelined Interests for better throughput

has helped us identify current NDN implementation shortcomings. We expect the testbed to be a valuable tool in exploring problems of the HEP domain and their NDN based solutions.

4. NDN for HEP: Opportunities and Challenges

Among the fields of data intensive science, the HEP community and those working on the LHC program in particular, make the largest scale use of network infrastructures and computing resources on a vast scale. This is a result of the inherently large size of the datasets that are produced by the experiments or by simulation, the global distribution of the sites where the data is produced, processed and/or analyzed by thousands of physicists, and the fact that the software base is under continuous development. An additional challenge is that the resources in all areas are not sufficient to rapidly accommodate all requests for data and/or results at any location. The bottlenecks are thus in multiple dimensions - network, storage and computing power. This necessitates the optimization of workflow in such a way that makes the most efficient use of the available resources, while serving the highest priority requests with a reasonably short turnaround time. Moreover, only optimizing the resources isn't enough, and steps must be taken to reduce wasting the resources and to reduce the turnaround times. An example of a current issue to be resolved is that of failed large transfers following completion of a long running job; it not only takes resources away from other competing requests, but it also leads to delays and inefficiencies in the research work of the entire scientific collaboration.

We believe several features of NDN can be beneficial to the HEP computing community. As an example, data sources such as CERN or any of the Tier1 sites can publish new content to the network following an agreed upon naming scheme. Data delivery is always performed in a pull mode, driven by a consumer at a Tier2 or Tier3 site issuing interest packets. When a data file or object collection is fetched, intermediate nodes in the network dynamically cache the data based on content popularity, and are thus ready to satisfy subsequent interests directly from the cache, lowering the load on servers with popular content. Combining this with the pull-mode results in a multicast-like data delivery, possibly optimizing both the network utilization as well as server load. Moreover, intelligent caching schemes can be deployed in the network to place data near the user.

Unlike IP, NDN offers the possibility of using multiple paths at the same time. Therefore, in case of a congested network, multiple data sources can be used to reduce dependency on any one path. There can also be multiple paths between a pair of nodes. The ability to use of multiple nodes and paths enables NDN to provide robust failover in case of network segment, node, or

end-site failure. Note that this failover is at the network layer and completely transparent to the applications or the users. This is in sharp contrast with current IP model where intelligence (such as tracking data replicas) for failover needs to be incorporated in the applications or middleware. Intelligence in the network layer makes NDN applications much simpler to develop and maintain.

All these are active research areas today. Caching as well as forwarding strategies, naming schemes, multi-sourcing and multi-path forwarding need to be investigated not only from the network but also the application perspective. HEP experiments using the Worldwide LHC Computing Grid (WLCG) have well-developed, hierarchical naming schemes in use, which already fit the NDN approach well. We take this logical file name structure as a starting point for investigating the benefits of using NDN as the data distribution and access network for HEP data processing. For this, we use the testbed described above. For the scalability study, we complement the testbed with the use of a simulation environment with a representative topology including network nodes and end-sites.

We further target simultaneous optimization of storage and bandwidth resource utilization through dynamic caching using the VIP framework first proposed in[9]. The VIP framework is the first comprehensive approach for addressing the fundamental and challenging problem of joint traffic engineering and caching in information-centric networks. The framework captures the important interactions among user demand patterns, network topology, and the locations of content sources and caches within a stochastic network setting. The VIP forwarding and caching algorithm presented in [9] achieves effective load balancing via both intelligent forwarding of content requests (i.e. Interest Packets), as well as appropriate cache selection and cache replacement. In wireline networks, the VIP algorithm can be implemented in a distributed manner. The combined forwarding and caching algorithm adaptively maximizes the user demand rate satisfied by the NDN network. Numerical experiments demonstrate the superior performance of the VIP algorithm in terms of low user delay, high rate of cache hits, and low rate of cache evictions, relative to classical routing and caching policies[9].

5. Effect of caching in simulated LHC network

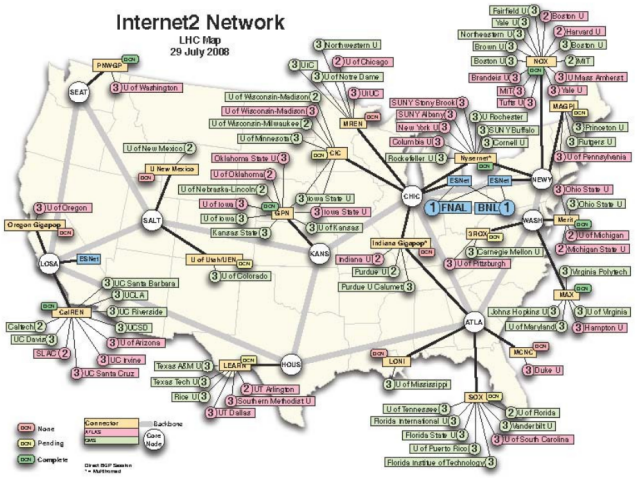

Figure 5. VIP simulation topology

In order to assess the potential performance gains of the NDN architecture and the VIP forwarding and caching framework[9], we carried out numerical simulations on the current LHC network in the U.S. The core of the network utilizes the Internet2 topology. Core link capacities

are set at 100 Gb/sec. Link capacities from exchange points to edge campus nodes are 10 Gb/sec, except those for ATLAS to University of Michigan, Indiana University and SLAC (100 Gb/sec), and for CMS to Caltech, Purdue, University of Nebraska-Lincoln, University of Wisconsin-Madison, University of Wisconsin-Milwaukee and University of Florida (100 Gb/sec), and to MIT (20 Gb/sec). The NDN architecture is used to transport requests and data for the LHC network. The Interest Packet size is set to 125 Bytes, while the Data Packet size (chunk size) is set to 200 MBytes. The content object size (file size) is 2 GBytes, the equivalent of 10 Data Packets (chunks). The VIP framework is used to optimize the forwarding of Interest Packets and the caching of Data Packets. Network state is maintained for the 20,000 most popular files (out of a total of 2 million files), which account for 40 percent of the requests. The data popularity distribution was fitted with a Zipf distribution, yielding a parameter of 0.8. Data requests (each consisting of 10 Interest Packets) arrive at each Tier2 or Tier3 node according to a Poisson process with rate λ (requests per second) where each arriving request is for data object i with probability p_i (independent of all other requests), where $\{p_i\}$ follows the Zipf(0.8) distribution. In this simulation, we considered the installation of 16 TByte cache nodes at the CHIC, NEWY, LOSA, and ATLA nodes in the Internet2 core, as well as the installation of 8 TByte cache nodes at Tier2 and Tier3 campus nodes. Figure 6 shows the average delay per data chunk (in seconds) versus the request arrival rate λ(in requests per second per site), for four different network scenarios: (1) NDN with VIP forwarding but no caching at either core or edge nodes; (2) NDN with VIP forwarding and caching at only the four core nodes; (3) NDN with VIP forwarding and caching at both the four core nodes and all Tier2 nodes; (4) NDN with VIP forwarding and caching at both the core nodes as well as all Tier2 and Tier3 edge nodes. It can be seen from Figure 6 that substantial reductions in the average request delay can be achieved through the use of caching, relative to the baseline performance with VIP forwarding. In particular, at a request arrival rate of 10 per second, the average delay per data chunk for NDN with VIP forwarding and caching at both the core and edge nodes is only 71% of the delay for NDN with VIP forwarding and no caching.

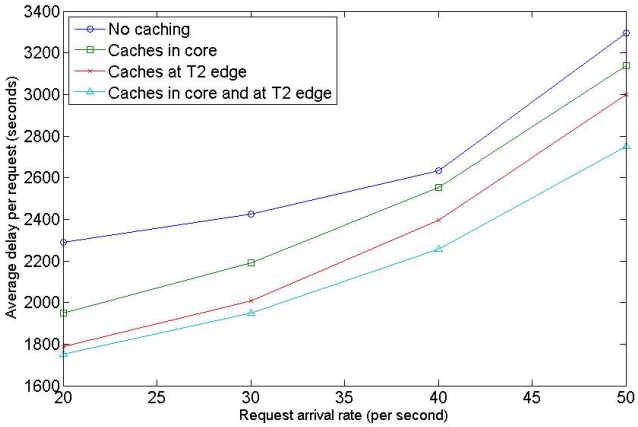

Figure 6. Delay performance of NDN/VIP-based LHC Network as function of request rate/ site.

Assuming for simplicity that the no caching case is roughly similar to the standard request for a file being satisfied by the remote server on each request, as would happen today in an TCP/IP network, the simulation results show the clear advantage of the NDN and VIP approach. Looking forward, we anticipate greater advantages of NDN caching in use cases with varying end-site connectivity and storage capacity, varying regional network loads, or where

the impact of temporarily impaired sectors of the network can be mitigated through adaptive caching strategies.

6. Future directions: Towards Data Intensive NDN

As in the case of the entry of HEP into the field of Grid Computing in the 1990's, the potential adoption of NDN by the HEP community requires an in-depth review of the underlying concepts and implementations in NDN, and the development of several new methods and concepts which need to be evolved through specialized simulations and field trials. There are a number of specific challenges to be met associated with moving datasets of Terabytes from globally distributed sites each storing from several to hundreds of petabytes, servicing requests to read and caching object collections of megabytes to a gigabyte at the rate of tens to hundreds of Hz, dealing with failed requests and/or redirection on the fly, and moderating the impact of these data operations on the world's research and education networks.

Meeting these challenges requires several areas of development including (a short list): (1) pervasive monitoring and tracking of tasks as well as the network state at each site and in each region, (2) a highly robust distributed set of repositories (catalogs) that can keep track of which datasets and collections have been requested recently and how often, (3) very rapid data transfers, from several to tens of Gbps each, to complete the largest data movements in a tolerable time, and in some cases to meet deadline schedules (4) the design, location, operation and management of the caches which are themselves significant storage facilities, (5) intelligent network operations and management of flows, including load balancing among alternate paths across complex multi-domain networks, and rate limiting of individual or aggregate flows to avoid congestion impeding other traffic (including non-HEP traffic) on any campus or in any constituent network.

The last area is synergistic with ongoing developments in the HEP and other communities, working with the major networks including ESNet, Internet2, GEANT and many other national and regional networks.

7. Acknowledgment

This work is supported in part by grants from DOE Offices of HEP and Advanced Scientific Computing (DE-SC0007346), NSF Grants (OCI-1341024, CNS-1205562, NSF- 1246133, NSF-13410999), and Cisco Research Grants (Micrograntt-2014-128271) to Caltech and Northeastern University. We thank Julian Bunn, Dorian Kcira and Samir Cury for their feedback and help.

8. References

[1] On-Demand Secure Circuits and Advance Reservation System. https://www.es.net/engineering-services/oscars/, 2015.
[2] http://monarc.web.cern.ch/MONARC. The monarc project, 2015.
[3] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.
[4] Gerald A. Meehl, Curt Covey, Bryant McAvaney, Mojib Latif, and Ronald J. Stouffer. Overview of the coupled model intercomparison project. *Bulletin of the American Meteorological Society*, 86(1):89–93, 2005.
[5] Catherine Olschanowsky, Susmit Shannigrahi, and Christos Papadopoulos. Supporting climate research using named data networking. In *Local & Metropolitan Area Networks (LANMAN), 2014 IEEE 20th International Workshop on*, pages 1–6. IEEE, 2014.
[6] Named Data Project on GitHub. https://github.com/named-data/, 2015.
[7] ESNet quick reference guide for Linux 2.6+ tuning. http://fasterdata.es.net/host-tuning/linux/, 2015.
[8] Karl E Taylor, V Balaji, Steve Hankin, Martin Juckes, Bryan Lawrence, and Stephen Pascoe. Cmip5 data reference syntax (drs) and controlled vocabularies, 2010.
[9] Edmund Yeh, Tracey Ho, Ying Cui, Michael Burd, Ran Liu, and Derek Leong. Vip: A framework for joint dynamic forwarding and caching in named data networks. In *Proceedings of the 1st international conference on Information-centric networking*, pages 117–126. ACM, 2014.